

APPLICATION FOR UNITED STATES LETTERS PATENT

For

CONTENT ADDRESSABLE MEMORY HAVING DYNAMIC MATCH
RESOLUTION

Inventor:

Varadarajan Srinivasan

Prepared by:

BLAKELY SOKOLOFF TAYLOR & ZAFMAN LLP
12400 Wilshire Boulevard
Los Angeles, CA 90025-1030
(408) 720-8300

Attorney's Docket No.:02489.P039C

"Express Mail" mailing label number: EV336590418US

Date of Deposit: 10/2/03

I hereby certify that I am causing this paper or fee to be deposited with the United States Postal Service "Express Mail Post Office to Addressee" service on the date indicated above and that this paper or fee has been addressed to the Commissioner for Patents, PO Box 1450, Alexandria, Virginia 22313-1450.

JUANITA BRISCOE

(Typed or printed name of person mailing paper or fee)

(Signature of person mailing paper or fee)

(Date signed)

CONTENT ADDRESSABLE MEMORY HAVING DYNAMIC MATCH RESOLUTION

Cross-reference to related application

[0001] This application is a Continuation, and claims benefit under 35 USC §120, of co-pending and commonly owned, U.S. Patent Application Serial No. 10/112,630 filed March 29, 2002 and entitled “Content Addressable Memory Having Dynamic Match Resolution,” which is incorporated by reference herein.

Field of Invention

[0002] The field of invention relates generally to the semiconductor arts; and, more specifically, to a method and apparatus for a content addressable memory (CAM) having dynamic match resolution.

Background

[0003] Content addressable memories (CAMs) can be used within a wealth of different electronic systems (e.g., computing systems such as laptop computers, personal computers (PCs), servers, etc.; networking systems such as routers, switches, base stations, etc.; etc.). Often, a CAM is used to perform a “look-up” function. For example, in networking applications, a portion of the header information of a packet is often used as a search key or comparand to look-up or identify an entry in the CAM. The address at which the matching entry is located in the CAM can then be used to address or index a location in associated memory where attributes as to how the packet is to be handled are found (e.g., an attribute that identifies the priority level of the packet, an

attribute that identifies the specific fiber-optic cable over which the packet is to be launched, etc.).

[0004] **Figure 1** shows an example of at least a portion 100 of a CAM architecture. According to the depiction of **Figure 1**, CAM architecture 100 includes a plurality (“K”) of rows of CAM cells 140₁ through 140_K. Each CAM cell includes a memory storage element and a compare circuit and can be binary (i.e., effectively stores and compares on two logic states - logic 0 and logic 1) or ternary (i.e., effectively stores and compares on three logic states - logic 0, logic 1, and a don’t care state).

[0005] A comparand is typically registered in comparand register 108 and then compared with the data values stored within the plurality of CAM cells 140. The comparison results are reflected as match signals on corresponding match signal lines 110₁-110_K. The match signals are typically then latched by latch 102. Latch 102 has K latch circuits 102₁-102_K that each latch a corresponding match signal state in response to the latch enable signal generated by timing generator 103. The latch enable signal is typically generated in response to a cycle clock and control signal CMP that indicates when a compare operation is taking place.

[0006] The latched match signals are provided to priority encoder 105 and flag logic 113 over signal lines 104₁-104_K. Priority encoder 105 processes the match signals in a predetermined manner to determine which of the matching entries (i.e., those data values in the CAM array that match the comparand) is the highest priority matching entry. An index or address of the highest priority matching entry in the CAM array is then output from priority encoder 105 over signal lines 106. Flag logic 113 typically generates a

match flag signal that indicates if one or more of the match signals indicates a match, and thus qualifies the output of the priority encoder.

[0007] Semiconductor integrated circuit (IC) designers typically specify minimum performance levels that the IC will meet or exceed across a wide range of operating conditions of a CAM device. The operating conditions include environmental conditions (e.g., temperature, moisture content in the operating environment, pressure, etc.), electrical conditions (e.g., supply voltage, input and output voltages, currents and loading, etc.), manufacturing variations (e.g., fabrication variations in transistor geometries and properties, material geometries and properties, etc., which are commonly referred to as “process variations”), and functional considerations (e.g., which operations have been performed by the device and which operations are currently being performed by the device, etc.).

[0008] With respect to CAM architecture 100, timing generator 103 is typically designed using simulation tools to transition the latch enable signal at a point in time when all of the match signals are stable after a compare operation. As such, the timing generator is typically designed to transition the latch enable signal to capture accurate match results even under worst case operating conditions. Worst case operating conditions are typically high operating temperature, low supply voltage, slow fabrication geometries, and the slowest time required for CAM array 140 to perform a match and/or a mis-match operation. Although this design approach may guarantee a minimum performance under worst case operating conditions, it sacrifices better performance that could have been realized under better than worst case operating conditions. This can adversely affect the rated performance of the CAM device and can decrease the overall

operating frequency, speed or throughput of the device. For example, the longer time that is required to latch the match signals, the later in a particular clock cycle (e.g., of the cycle clock) that it takes for the priority encoder and flag logic to perform their respective operations. This slows down the maximum operating speed of the device.

[0009] Additionally, timing generator 103 is typically designed as a pulse generator (e.g., a one-shot) and/or a delay circuit (e.g., one or more inverting or non-inverting buffer circuits or resistor-capacitor (RC) circuits). Since these circuits are designed differently than CAM array 140 (i.e., do not include CAM cells), as operating conditions change the latch enable signal output by timing generator 103 typically does not track in time or voltage proportionally with changes to the match results provided on match lines 110. This variation is further simulated and accounted for during design, and typically results in additional delay added to the point in time at which the latch enable signal is designed to transition.

Figures

[0010] The present invention is illustrated by way of example, and not limitation, in the Figures of the accompanying drawings in which:

[0011] **Figure 1** shows an example of a portion of a CAM architecture.

[0012] **Figure 2** shows an embodiment of a portion of a CAM architecture including a dynamic timing generator.

[0013] **Figure 3** shows one embodiment of a row of CAM cells.

[0014] **Figure 4** shows an exemplary timing diagram of the operation of the row of CAM cells of **Figure 3**.

[0015] **Figure 5** shows an embodiment of a binary CAM cell.

[0016] **Figure 6** shows another embodiment of a row of CAM cells.

[0017] **Figure 7** shows an exemplary timing diagram of the operation of the row of CAM cells of **Figure 6**.

[0018] **Figure 8** shows an exemplary timing diagram of the dynamic operation of the dynamic timing generator of **Figure 2**.

[0019] **Figure 9** illustrates the operation of the CAM architecture of **Figure 2**.

[0020] **Figure 10** shows an embodiment of the dynamic timing generator of **Figure 2**.

[0021] **Figure 11** shows an embodiment of the match or mis-match comparison timing circuits of **Figure 10**.

[0022] **Figure 12** shows an embodiment of the operation of the comparison timing circuit of **Figure 11**.

[0023] **Figure 13** shows another embodiment of the match or mis-match comparison timing circuits of **Figure 10**.

[0024] **Figure 14** shows another embodiment of the match or mis-match comparison timing circuits of **Figure 10**.

[0025] **Figure 15** shows another embodiment of the match or mis-match comparison timing circuits of **Figure 10**.

[0026] **Figure 16** shows an embodiment of a CAM device that includes the CAM architecture of **Figure 2**.

[0027] **Figure 17** shows an embodiment of an application that includes the CAM device of **Figure 16**.

Description

[0028] In the following description, for purposes of explanation, specific nomenclature is set forth to provide a thorough understanding of the present invention. However, it will be apparent to one skilled in the art that these specific details may not be required to practice the present invention. In some instances, the interconnection between circuit elements or blocks may be shown as buses or as single signal lines. Each of the buses may alternatively be single signal lines, and each of the single signal lines may alternatively be buses. A signal is said to be “asserted” when the signal is driven to a low or high logic state (or charged to a high logic state or discharged to a low logic state) to indicate a particular condition. Conversely, a signal is said to be “deasserted” to indicate that the signal is driven (or charged or discharged) to a state other than the asserted state (including a high or low logic state, or the floating state that may occur when the signal driving circuit is transitioned to a high impedance condition, such as an open drain or open collector condition). A signal driving circuit is said to “output” a signal to a signal receiving circuit when the signal driving circuit asserts (or deasserts, if explicitly stated or indicated by context) the signal on a signal line coupled between the signal driving and signal receiving circuits. A signal line is said to be “activated” when a signal is asserted on the signal line, and “deactivated” when the signal is deasserted. Additionally, the prefix symbol “/” attached to signal names indicates that the signal is an active low signal (i.e., the asserted state is a logic low state). A line over a signal name (e.g., $\overline{\text{< signal name >}}$) is also used to indicate an active low signal. Active low signals may be changed to active high signals and vice-versa as is generally known in the art.

Additionally, the term “coupled” as used herein refers to the direct or indirect connection through one or more intervening circuits.

[0029] A CAM architecture is disclosed that dynamically adjusts its match resolution timing in response to the applicable operating conditions under which the CAM architecture operates such that the CAM architecture can exhibit improved operating performance (e.g., operating speed, throughput, etc.) during a compare operation relative to a CAM architecture designed according to traditional approaches. The CAM architecture includes a dynamic timing generator that performs one or more compare operations to parallel the operation of a CAM array. Thus, the CAM architecture employs the dynamic timing generator to “self-time” capture of match results from the CAM array. The dynamic timing generator outputs an enable signal to capture the match results from the CAM array, and dynamically adjusts the output of the enable signal as operating conditions change (e.g., a change in temperature or a degradation in transistor gain).

[0030] For example, if operating conditions improve such that faster compare operations are realized by the CAM array, the dynamic timing generator adjusts its enable signal such that match results from the CAM array are captured earlier in time than under relatively slower operating conditions. Likewise, as operating conditions degrade such that slower compare operations are realized by the CAM array, the dynamic timing generator adjusts its enable signal such that match results from the CAM array are captured later in time than under relatively faster operating conditions. As such, the match results can be captured as soon as they are available from the CAM array.

[0031] **Figure 2** shows an embodiment of at least a portion of a CAM architecture 200 having dynamically adjustable match resolution as described just above. CAM

architecture 200 includes comparand register 208, CAM array 240, latch 202, priority encoder 205, flag logic 213, and dynamic timing generator 220. The CAM architecture may include other circuits (e.g., as shown in **Figure 16**) including, for example, a control circuit or instruction decoder that decodes instructions (e.g., compare instructions) and provides control signals to one or more of the circuit blocks, read circuitry and write circuitry for the CAM array, address circuitry, configurability circuits to configure the width and/or depth of the CAM array, error detection and/or correction circuitry, and the like.

[0032] CAM array 240 includes K of rows of CAM cells 240₁-240_K. Each CAM cell includes at least one memory storage element and at least one compare circuit and can be binary (i.e., effectively stores and compares on two logic states - logic 0 and logic 1) or ternary (i.e., effectively stores and compares on three logic states - logic 0, logic 1, and a don't care state). A comparand is typically registered in comparand register 208 and then compared with the data values stored within the plurality of CAM cells 240. Comparand register 208 is optional. The comparison results are reflected as match signals on corresponding match signal lines 210₁-210_K. The match signals are typically then latched by latch 202. Latch 202 has K latch circuits 202₁-202_K that each latch a corresponding match signal state in response to the latch enable signal 203 generated by dynamic timing generator 220. The latch enable signal 203 is typically generated in response to a cycle clock, and may also be generated in response to a control signal that indicates when a compare operation is taking place. The control signal may be generated, for example, by a control circuit or instruction decoder that detects when a compare operation is to take

place on the CAM array. Latch 202 may also be any type of timed or clocked storage circuit including, for example, a register (e.g., a flip-flop) or other timed memory circuit.

[0033] The latched match signals are provided to priority encoder 205 and flag logic 213 over signal lines 204₁-204_K. Priority encoder 205 processes the match signals in a predetermined manner to determine which of the matching entries (i.e., those data values in the CAM array that match the comparand) is the highest priority matching entry. An index or address of the highest priority matching entry in the CAM array is then output from priority encoder 205 over signal lines 206. For other embodiments, the priority encoder may be an encoder and not a priority encoder.

[0034] Flag logic 213 generates a match flag signal that indicates if one or more of the match signals indicates a match, and thus qualifies the output of the priority encoder. The flag logic may also indicate the fill state of the CAM array by, for example, outputting a full flag signal, an empty signal or a partially full flag signal (e.g., by monitoring one or more valid bits each associated with an entry in the CAM array).

[0035] **Figure 3** shows a row of CAM cells 340 that is one example of a row of CAM cells 240 of **Figure 2**. Row 340 includes pre-charge circuit 320 and N CAM cells 330. Pre-charge circuit 320 is any type of pre-charge circuit that charges match line 310 towards the power supply rail VDD. Each of the N CAM cells may be binary, ternary or other types of CAM cells, that each include at least one memory storage element for storing data and at least one compare circuit for comparing at least bit of the comparand with the stored bit in the CAM cell. Row 340 will be generally referred to hereafter as “NOR-based” because if any compare circuit within a particular CAM cell determines that its stored data bit does not match a corresponding bit in the comparand, then the

match line is discharged by the CAM cell towards ground. Only if all of the CAM cells store data that matches corresponding bits in the comparand will the match line remain in a pre-charged state.

[0036] This operation is further shown in **Figure 4** in which from time T0 to time T1 there is a match state. At time T1 there is at least one CAM cell that stores a bit that mismatches a corresponding bit in the comparand, and that CAM cell starts to discharge the match line towards ground. At time T2, the voltage on match line 310 crosses VTH, which is the voltage that delineates a match from a mis-match state on the match line. At time T3, a match condition is again determined. At time T4 the match condition is represented on match line 310.

[0037] **Figure 5** shows binary CAM cell 530 that is one embodiment of any of CAM cells 330 of **Figure 3**. CAM cell 530 includes a memory cell 540 and a compare circuit 550. Memory cell 540 may be any type of memory cell including a SRAM cell, DRAM, cell or any other type of volatile or non-volatile memory cell. As shown, both the data bit and its logical complement stored in memory cell 540 are provided to respective gates of transistor Q4 and Q2. Transistor Q2 is coupled in series with transistor Q1 between the match line and ground. Transistor Q1 has its gate coupled to a bit of comparand data C. Similarly, transistor Q4 is coupled in series with transistor Q3 between the match line and ground. Transistor Q4 has its gate coupled to receive the logical complement of the comparand bit. As shown in **Figure 5**, if the stored data and the comparand are the same logical state (i.e., match), then no path to ground is established for the match line. If, however, they do not have the same logical state (i.e., do not match), then a discharge

path to ground exists for the match line. Other embodiments of compare circuit may be used.

[0038] **Figure 6** shows a row of CAM cells 640 that is another embodiment of any of the row of CAM cells of **Figure 2**. Row 640 includes pre-charge circuit 320, N CAM cells 630₁-630_N, and N pass transistors Q1-QN. Each of the N CAM cells 630₁-630_N may be binary, ternary or other types of CAM cells, that each include at least one memory storage element for storing data and at least one compare circuit for comparing at least one bit of the comparand with the stored bit in the CAM cell. Row 640 will be referred to generally hereafter as “NAND-based” because the match line is discharged toward ground only if all of the compare circuits within the CAM cells determine that its stored data matches the comparand. When a CAM cell stores data that matches a corresponding bit in the comparand, it sends a signal to the gate of its corresponding pass transistor to turn on the transistor. If any one of the CAM cells stores data that does not match a corresponding bit in the comparand, the corresponding pass transistor is turned off and the match line remains pre-charged towards VDD.

[0039] This operation is further shown in **Figure 7** in which from time T0 to time T1 there is a mis-match state. At time T1 all of stored bits match the comparand such that all of the pass transistors are enabled to discharge the match lines towards ground. At time T2, the voltage on match line 310 crosses V_{TH}, which is the voltage that delineates a match from a mis-match state on the match line. At time T3, a mis-match condition is again determined. At time T4 the mis-match condition is represented on match line 610.

[0040] Returning again to **Figure 2**, dynamic timing generator 220 transitions the latch enable signal to cause latch 202 to capture the match results when all of the match

signals accurately reflect the current compare results. In contrast to timing generator 103 of **Figure 1**, dynamic timing generator 220 performs one or more compare operations to parallel the operation of CAM array 240. Thus, as the generation of the match signals is increased or decreased in time by the operation of CAM array 240 under various operating conditions, the dynamic timing generator similarly and proportionally dynamically adjusts the time at which it transitions the latch enable signal. As such, CAM architecture 200 “self-times” when to capture the match results enabling them to be captured as soon as they are stable on the match lines.

[0041] For example, because dynamic timing generator 220 performs a parallel comparison operation as that performed by CAM array 240, if operating conditions improve such that faster compare operations are realized by the CAM array, the dynamic timing generator adjusts the transition of the latch enable signal by a substantially proportional amount such that match results from the CAM array are captured earlier in time than under relatively slower operating conditions. This may allow the CAM architecture to run at higher frequencies under faster operating conditions. Likewise, as operating conditions degrade such that slower compare operations are realized by the CAM array, the dynamic timing generator adjusts the transition of the latch enable signal by a substantially proportional amount such that match results from the CAM array are captured later in time than under relatively faster operating conditions.

[0042] For example, as shown in **Figure 8**, if a new comparand begins to be analyzed on the rising edge of a cycle clock signal 207, and the match results are latched on the rising edge of the latch enable signal, then using the rising edges 811 and 813 of the cycle clock 207, respectively, as a reference point for both latch times 812 and 814, note that

the first latch time 812 is observed after time period Tx for a first compare cycle, and the second latch time 814 is observed as time period Ty for a second compare cycle. Note that time period Ty is longer in time than time period Tx. The second compare cycle represents operating conditions that cause CAM architecture 200 to operate slower (i.e., generate accurate match results slower) than in the first compare cycle. This difference in time periods demonstrates that the latch time of the CAM architecture can be dynamically adjusted. Furthermore, in at least one embodiment, a new latch time can be established with each new comparand. As such, the CAM approach of **Figure 2** is able to vary the latch timing on a comparand-by-comparand basis.

[0043] By having dynamic timing generator 220 perform a substantially similar compare operation as that performed by CAM array 240, the dynamic timing generator can be designed to transition the latch enable signal at a time that more quickly (i.e., as opposed to conventional simulation designs) indicates when the match signals accurately reflect the match results from the CAM array for a given compare operation. This advantageously improves the speed at which the match results are latched and when the priority encoder can output a matching index, and when the flag logic can output a match result. Consequently, this can improve the operating frequency, speed and throughput of the CAM architecture over conventional designs.

[0044] **Figure 9** illustrates one embodiment of the operation dynamic timing generator 220 and latch 202. For this embodiment, 900 and 901 are performed by dynamic timing generator 220, and 902 is performed by latch 202. The amount of time that the CAM array 204 uses to determine its match results on match will be referred to hereafter as a “comparison time”, a “compare time” or the like.

[0045] At 900, while a comparison operation is being performed by the CAM array, dynamic timing generator 220 determines the longest or worst case comparison time that the CAM array should experience for the given operating conditions. For one example, dynamic timing generator 220 determines the longest comparison time by performing its own parallel comparison operations (with the supplied comparand or with a predetermined, pre-stored comparand) to mirror the operation of the CAM array. The parallel comparison operations may include comparison operations that generate both match and mis-match results so as to mirror the slowest functional operation that any row of the CAM array could experience in a given compare operation.

[0046] Once dynamic timing generator 220 determines the longest comparison time, the dynamic timing generator transitions the latch enable signal at 901 such that the match results on match signal lines 210 are captured by latch 202 (at 902) at or beyond the just determined longest comparison time. As a result, the match results are latched as soon as practicable under the applicable operating conditions. This can avoid idling for significant amounts of time prior to latching match results from the CAM array once they are ready to be latched.

[0047] Note that the process of **Figure 9** can be continually repeated so that an appropriate latch time is continually re-determined over the course of operation of the CAM architecture. As such, as described above, the CAM can dynamically adjust to changes in operating conditions that arise during its operation. In one embodiment, the process is continually repeated to generate a new latch enable signal in every clock cycle. Alternatively, a new latch enable signal may be generated at a frequency slower than every clock cycle.

[0048] It is important to point out that numerous embodiments may differ with respect to the manner in which dynamic timing generator 220 establishes the latch timing for latch 202. For example, according to at least one approach, a plurality of comparison time measurements are undertaken in order to monitor a plurality of circumstances that could noticeably affect the compare operation of CAM array 240.

[0049] **Figure 10** illustrates dynamic timing generator 1020 that is one embodiment of dynamic timing generator 220 of **Figure 2**. Dynamic timing generator 1020 includes match comparison timing circuit 1001, mis-match comparison timing circuit 1006, and select circuit 1005. Each of the comparison timing circuits 1001 and 1006 performs compare operations to mirror or parallel the similar worst-case match and mis-match comparison operations performed by CAM array 240. Various timing circuit designs may be used to perform the parallel comparison operations including using one or more duplicate rows of CAM cells such as those used in the CAM array.

[0050] Each comparison timing circuit 1001 and 1006 outputs a comparison result signal at respective outputs 1002 and 1004. The comparison result signals are indicative of the time required to complete the corresponding match or mis-match operation. Select circuit 1005 selects the slower of the comparison result signals to determine when to transition the latch enable signal 203. That is, select circuit 1005 effectively recognizes which of the match and mis-match comparison times performed by timing circuits 1001 and 1006 represents the slowest comparison time. This slowest comparison time is also indicative of the slowest possible comparison time that any given row of the CAM array may take to resolve its own match results.

[0051] Dynamic timing generator 1020 uses comparison timing circuits 1001 and 1006 to mirror the two operations and results (i.e., match and mis-match compare operations) that a given row of CAM cells in the CAM may experience during a compare operation. For one embodiment, match comparison timing circuit 1001 includes a row of CAM cells or compare circuits (manufactured in a similar manner to, and including substantially similar circuit components to, those in the CAM array) that performs a compare operation with a pre-stored comparand such that the comparison always generates a match (e.g., as a match signal on output 1002). Similarly, mis-match comparison timing generator circuit 1006 may include a row of CAM cells or compare circuits (manufactured in a similar manner as to, and including substantially similar circuit components to, those in the CAM array) that performs a compare operation with a pre-stored comparand such that the comparison always generates a mis-match (e.g., as a signal on output 1002). Select circuit 1005 compares or combines the timing or logic states to select the slowest results reflected on lines 1002 and 1004.

[0052] For one embodiment, each of the comparison result signals is an active high signal that is asserted to the active high state when the respective timing circuit has completed its compare function(s). For this embodiment, select circuit 1005 may be a logic AND gate.

[0053] Two compare timing circuits 1001 and 1006 are included in dynamic timing generator 1020 because either a match or mis-match comparison result for a given row of CAM cells in the CAM array may take longer. If, either a match or mis-match will always take longer, than only one such comparison timing generator circuit need be included in dynamic timing circuit 1020. Alternatively, if a row of CAM cells in the

CAM array can have more than one state, then additional timing circuits may be included in dynamic timing generator 1020, one for each possible output state of the rows of CAM cells of the CAM array.

[0054] Note that match comparison timing circuit 1001 can determine the worst-case time that it would take for a match comparison operation to take place in any row of CAM array 240. This may be accomplished, for example, by performing a series of comparison operations to precondition output line 1002 to a particular state before generating a match condition. Exemplary embodiments will be described below.

Similarly, mis-match comparison timing circuit 1006 can determine the worst-case time that it would take for a mis-match comparison operation to take place in any row of CAM array 240. This may be accomplished, for example, by performing a series of comparison operations to precondition output line 1004 to a particular state before generating a mis-match condition. Exemplary embodiments will be described below.

[0055] Thus, dynamic timing generator 1020 may be tailored so as to test against both a worst case match comparison condition and a worst case mis-match comparison condition. In this sense, the design approach of **Figure 10** is comprehensive because a plurality of potential worst case circumstances are individually measured or determined, and the worst result across the plurality of worst case circumstances is used to set the CAM's latch time. As such, the probability of a race condition occurring is low because the latch time is set at or beyond what is believed to be the slowest possible CAM comparison time.

[0056] Note that since timing circuits 1001 and 1006 perform the same functions as the rows of CAM cells in CAM array 204, that as operating conditions change the

comparison results generated by timing circuits 1001 and 1006 adjust in a similar manner (e.g., by a similar or proportional amount of time, voltage, etc.) as the match results generated by CAM array 240. When the timing circuits 1001 and 1006 are formed using similar circuits and geometries (e.g., also use rows of similarly designed CAM cells or compare circuits to those used in CAM array 240), changes in operating conditions substantially affect both the CAM array and the timing circuits in substantially the same way, thus allowing the benefits of on-chip latch time determination to dynamically track with changes in operating conditions.

[0057] Figure 11 illustrates comparison timing circuit 1100 that is one embodiment of the timing circuits 1001 and 1006 of **Figure 10**. Comparison timing circuit 1100 includes storage circuit 1102, compare circuit 1104, select circuit 1106, and sense circuit 1108. Compare circuit 1104 compares the data stored in storage circuit 1102 with either comparand A (CMPDA) or comparand B (CMPDB) provided by select circuit 1106. CMPDA and CMPDB may be stored in storage circuits or may be provided external to the CAM architecture.

[0058] Select circuit 1106 determines which comparand to supply to compare circuit 1104 in response to the comparand select signal CMPDSEL. CMPDSEL may be generated, for example, by a control circuit or instruction decoder, a timing circuit, or may be free running. Select circuit provides its output to the compare circuit also in response to the cycle clock. For one embodiment, select circuit 1106 is a multiplexer. The result of the comparison operation performed by compare circuit 1104 is provided on signal line 1110 and sensed by sensing circuit 1108. Sensing circuit 1108 senses the voltage change of the comparison result on signal line 1110 and detects when the voltage

has reached one or more thresholds. Sensing circuit 1108 then asserts or deasserts a signal on signal line 1112. Sensing circuit 1108 may be any type of sensing circuit including a single or differential sense amplifier, a logic gate (e.g., an inverter, AND gate, OR gate, etc.) and the like. The sensed output is provided to signal line 1112 that corresponds to comparison result outputs 1002 and 1004 of **Figure 10**.

[0059] For one embodiment, storage circuit 1102 and compare circuit 1104 form a row of CAM cells (e.g., such as that shown in **Figures 3 or 6**), and signal line 1110 corresponds to a match line. To mirror and parallel the operation of the rows of CAM cells in the CAM array 240 of **Figure 2**, the row of CAM cells formed by storage circuit 1102 and compare circuit 1104 are manufactured in a similar manner to, and include substantially similar circuit components to, those in the CAM array.

[0060] Note that some form of deliberate additional delay may be added to the output of comparison timing circuit 1100 (e.g., via the propagation of one or more logic gates) to ensure that a latch time which is derived from the result is beyond the comparison times of the individual rows of CAM cells in CAM array 240. By so doing, race conditions can be avoided.

[0061] For alternative embodiments, data stored in storage circuit 1102 may be hard-wired to particular logic states (i.e., not stored in memory circuits such as those found in CAM array 240) to reduce the size of storage circuit. This can be accomplished while still maintaining substantially similar responses between the comparison timing circuit 1100 and the rows of CAM cells in CAM array 240 during compare operations across varying operating conditions.

[0062] In still other embodiments, storage circuit 1102 and compare circuit 1104 may be replaced by a circuit whose timing characteristics (e.g., its RC time constant) are designed to correlate to the comparison timing characteristics of the rows of CAM cells in CAM array 240 (i.e., substantially tracks the voltage changes on the match lines of the CAM array in response to changes in operating conditions). Because of the design for correlation, the output from such a comparison timing circuit can still be used to set the CAM's latch time even if a comparison is not actually performed by the timing circuit. Again, those of ordinary skill will be able to determine the appropriate correlation and/or any additional delay to avoid race conditions.

[0063] The operation of comparison timing circuit 1100 is further illustrated in **Figure 12**. At 1202, CMPDSEL is in a first state such that select circuit 1106 provides CMPDA to compare circuit 1104. Compare circuit 1104 compares CMPDA with the data stored in storage circuit 1102 as many times as is required to pre-condition the signal line 1110 to a first state. Subsequently, at 1204, CPMDSEL transitions to a second state such that select circuit 1106 provides CMPDB to compare circuit 1106. Compare circuit 1106 compares CMPDB with the data stored in storage circuit 1102, and sense circuit 1108 senses when the voltage on signal line 1110 has reached one or more thresholds.

[0064] Consider when comparison timing circuit 1100 is operating as a match comparison timing circuit such as 1001 of **Figure 10**. Comparison timing circuit 1100 determines the worst-case (i.e., slowest) comparison time of any row of CAM cells in CAM array 240 to determine a match condition under current operating conditions by having CMPDA set to mis-match the data stored in storage circuit 1102 and CMPDB set to match the data stored in storage circuit 1102. For example, when CMPDA mis-

matches the data stored in storage circuit 1102 (at 1202), the voltage on signal line 1110 will be discharged towards ground potential (for a NOR-based row of CAM cells). This may take one or more clock cycles to discharge signal line 1110 as close to ground potential as possible, but can be more quickly achieved by having CMPDA mis-match all bits of the data stored in storage circuit 1102. Then, CMPDB is provided at 1204 and sensing circuit 1108 senses when the signal on signal line 1110 is equal to or greater than V_{TH} (e.g., as shown in **Figure 4**). Note that this example has been discussed with reference to NOR-based CAM cell operation, but can be readily adapted to NAND-based CAM cell operation.

[0065] Now consider when comparison timing circuit 1100 is operating as a mis-match comparison timing circuit such as 1006 of **Figure 10**. Comparison timing circuit 1100 determines the worst-case (i.e., slowest) comparison time of any row of CAM cells in CAM array 240 to determine a mis-match condition under current operating conditions by having CMPDA set to match the data stored in storage circuit 1102 and CMPDB set to mis-match the data stored in storage circuit 1102. For example, when CMPDA matches the data stored in storage circuit 1102 (at 1202), the voltage on signal line 1110 will be charged towards the power supply rail VDD (for a NOR-based row of CAM cells). This may take one or more clock cycles to charge signal line 1110 as close to VDD as possible. Then, CMPDB is provided at 1204 and sensing circuit 1108 senses when the signal on signal line 1110 is equal to or less than V_{TH} (e.g., as shown in **Figure 4**). The worst case mis-match pattern (i.e., the one that will cause the signal on signal line 1110 to take the longest time to discharge towards V_{TH} and ground) is when only a single-bit of CMPDB mis-matches a corresponding single bit in the data stored in storage circuit 1002

because only a single CAM cell will be discharging signal line 1110. So, the worst-case mis-match pattern may be used to more closely parallel worst-case mis-match operations that may take place in any row of CAM cells in CAM array 240. Note that this example has been discussed with reference to NOR-based CAM cell operation, but can be readily adapted to NAND-based CAM cell operation.

[0066] Note that comparison timing circuit 1100 of **Figure 11** generates a compare results on signal line 1110 and 1112 every cycle clock. However, it generates a worst-case match or mis-match compare result at a frequency less than every cycle clock.

Figure 13 illustrates a comparison timing circuit 1300 that is another embodiment of either of timing circuits 1001 or 1006 of **Figure 10**. Comparison timing circuit 1300 includes interleaver circuit 1302 and M comparison timing circuits 1100 as shown in **Figure 11**, where M is any number. Each of the comparison timing circuits generates a worst-case match or mis-match comparison result as described above, except that the CMPDSEL signal for the respective comparison timing circuits 1100 are each set to a logic one state once every M clock cycles, and each CMPDSEL is set to a logic one state in a different clock cycle such that for any given cycle of the cycle clock, one of the circuits 1000 is outputting a worst-case match or mis-match comparison result (depending on whether it is configured to operate as 1001 or 1006 of **Figure 10**).

Interleaver circuit 1302 combines the outputs of the timing circuits 1100, and provides the combined output to signal line 1304 that corresponds to outputs 1002 or 1004 of **Figure 10**. For one embodiment, interleaver circuit 1302 is an OR logic gate. For other embodiments, interleaver circuit 1302 may be one or more other logic gates or circuits.

[0067] **Figure 14** illustrates the operation of comparison timing generator 1300 of **Figure 13** using two comparison timing generator circuits 1100. Consider, for this example, that comparison timing circuit 1300 operates as match comparison timing generator circuit 1001 of **Figure 10**. In a first cycle of the cycle clock, $CMPDSEL_1$ is set to a logic one state to select CMPDA which has all of its bits set to mis-match the data stored in each of the storage circuits in the timing circuits, and $CMPDSEL_2$ is set to a logic zero state to select CMPDB which has all of its bits set to match the data stored each of the storage circuits in the timing circuits. In the following second cycle of the cycle clock, $CMPDSEL_1$ is set to a logic zero state to select CMPDB and $CMPDSEL_2$ is set to a logic one state to select CMPDA, and the sense circuit of timing circuit 1100₁ asserts the signal on signal line 1110₁ when a match is detected (and deasserts the signal on signal line 1110₁ before the next match result is determined by 1100₂). In the following third cycle of the cycle clock, $CMPDSEL_1$ is again set to a logic one state to select CMPDA, and $CMPDSEL_2$ is again set to a logic one state to select CMPDB, and the sense circuit of timing circuit 1100₂ asserts the signal on signal line 1110₂ when a match is detected (and deasserts the signal on signal line 1110₂ before the next match result is determined by 1100₁). In this manner, comparison timing circuit 1300 of **Figure 14** outputs a worst-case match comparison timing signal on signal line 1304 every cycle of the cycle clock for use in latching match results from CAM array 240 of **Figure 2**.

[0068] Similarly, consider another example in which comparison timing circuit 1300 of **Figure 13** operates as match comparison timing generator circuit 1006 of **Figure 10**. In a first cycle of the cycle clock, $CMPDSEL_1$ is set to a logic one state to select CMPDA which has all of its bits set to match the data stored in each of the storage circuits in the

timing circuits 1100, and CMPDSEL₂ is set to a logic zero state to select CMPDB which has, for example, one of its bits set to mis-match a corresponding data bit stored in each of the storage circuits in the timing circuits. In the following second cycle of the cycle clock, CMPDSEL₁ is set to a logic zero state to select CMPDB and CMPDSEL₂ is set to a logic one state to select CMPDA, and the sense circuit of timing circuit 1100₁ asserts the signal on signal line 1110₁ when a mis-match is detected (and deasserts the signal on signal line 1110₁ before the next mis-match result is determined by 1100₂). In the following third cycle of the cycle clock, CMPDSEL₁ is again set to a logic one state to select CMPDA, and CMPDSEL₂ is again set to a logic one state to select CMPDB, and the sense circuit of timing circuit 1100₂ asserts the signal on signal line 1110₂ when a mis-match is detected (and deasserts the signal on signal line 1110₂ before the next mis-match result is determined by 1100₁). In this manner, comparison timing circuit 1300 of **Figure 14** outputs a worst-case mis-match comparison timing signal on signal line 1304 every cycle of the cycle clock for use in latching match results from CAM array 240 of **Figure 2**.

[0069] **Figure 15** illustrates the operation of comparison timing generator 1300 of **Figure 13** using three comparison timing generator circuits 1100. Consider, for this example, that comparison timing circuit 1300 operates as match comparison timing generator circuit 1001 of **Figure 10**. In a first cycle of the cycle clock, CMPDSEL₁ and CMPDSEL₂ are each set to a logic zero state to select CMPDA which has all of its bits set to mis-match the data stored in each of the storage circuits in the timing circuits 1100, and CMPDSEL₃ is set to a logic zero state to select CMPDB which has all of its bits set to match the data stored in each of the storage circuits in the timing circuits. In the

following second cycle of the cycle clock, CMPDSEL₂ and CMPDSEL₃ are each set to a logic zero state to select CMPDB, and CMPDSEL₁ is set to a logic one state to select CMPDA, and the sense circuit of timing circuit 1100₁ asserts the signal on signal line 1110₁ when a match is detected (and deasserts the signal on signal line 1110₁ before the next match result is determined by 1100₂). In the following third cycle of the cycle clock, CMPDSEL₁ and CMPDSEL₃ are each set to a logic zero state to select CMPDB, and CMPDSEL₂ is set to a logic one state to select CMPDA, and the sense circuit of timing circuit 1100₂ asserts the signal on signal line 1110₂ when a match is detected (and deasserts the signal on signal line 1110₂ before the next match result is determined by 1100₃). In the following fourth cycle of the cycle clock, CMPDSEL₁ and CMPDSEL₂ are each again set to a logic zero state to select CMPDB, and CMPDSEL₃ is again set to a logic one state to select CMPDA, and the sense circuit of timing circuit 1100₃ asserts the signal on signal line 1110₃ when a match is detected (and deasserts the signal on signal line 1110₃ before the next match result is determined by 1100₁). In this manner, comparison timing circuit 1300 of **Figure 15** outputs a worst-case match comparison timing signal on signal line 1304 every cycle of the cycle clock for use in latching match results from CAM array 240 of **Figure 2**.

[0070] **Figure 15** illustrates the operation of comparison timing generator 1300 of **Figure 13** using three comparison timing generator circuits 1100. Consider, for this example, that comparison timing circuit 1300 operates as a mis-match comparison timing generator circuit 1006 of **Figure 10**. In a first cycle of the cycle clock, CMPDSEL₁ and CMPDSEL₂ are each set to a logic one state to select CMPDA which has all of its bits set to match the data stored in each of the storage circuits in the timing circuits 1100, and

CMPDSEL₃ is set to a logic zero state to select CMPDB which has, for example, a single one of its bits set to mis-match a corresponding data bit stored in each of the storage circuits in the timing circuits. In the following second cycle of the cycle clock, CMPDSEL₂ and CMPDSEL₃ are each set to a logic zero state to select CMPDB, and CMPDSEL₁ is set to a logic one state to select CMPDA, and the sense circuit of timing circuit 1100₁ asserts the signal on signal line 1110₁ when a mis-match is detected (and deasserts the signal on signal line 1110₁ before the next mis-match result is determined by 1100₂). In the following third cycle of the cycle clock, CMPDSEL₁ and CMPDSEL₃ are each set to a logic zero state to select CMPDB, and CMPDSEL₂ is set to a logic one state to select CMPDA, and the sense circuit of timing circuit 1100₂ asserts the signal on signal line 1110₂ when a mis-match is detected (and deasserts the signal on signal line 1110₂ before the next mis-match result is determined by 1100₃). In the following fourth cycle of the cycle clock, CMPDSEL₁ and CMPDSEL₂ are each again set to a logic zero state to select CMPDB, and CMPDSEL₃ is again set to a logic one state to select CMPDA, and the sense circuit of timing circuit 1100₃ asserts the signal on signal line 1110₃ when a mis-match is detected (and deasserts the signal on signal line 1110₃ before the next mis-match result is determined by 1100₁). In this manner, comparison timing circuit 1300 of **Figure 15** outputs a worst-case mis-match comparison timing signal on signal line 1304 every cycle of the cycle clock for use in latching match results from CAM array 240 of **Figure 2**.

[0071] Note that the match comparison timing circuit 1001 and the mis-match comparison timing circuit 1006 of **Figure 10** may each have a different number of timing circuits 1100. For example, the match comparison timing circuit may have two timing

circuits such as shown in **Figure 14**, and the mis-match comparison timing circuit may have three timing circuits such as shown in **Figure 15**.

[0072] **Figure 16** shows 16CAM apparatus or device 1600 that is one embodiment of a CAM device that includes CAM architecture 200 of **Figure 2**. The CAM apparatus may be formed, for example, as an integrated circuit on a semiconductor substrate.

[0073] CAM device 1600 includes addressing unit 1610 that is capable of addressing one or more rows of CAM cells in CAM array 240 (or one or more individual CAM cells). Addressing unit 1610 may include, for example, a decoder that is coupled to word lines in the CAM array. The addresses are provided along an address bus 1617.

[0074] The Read/Write unit 1611 is capable of reading and writing data words from/to one or more rows of CAM cells (or one or more individual CAM cells) in CAM array 240. For example, Read/Write unit 1611 may include one or more sense amplifiers for sensing and amplifying data stored in CAM array 240 during a read operation, and one or more write drivers for driving data to one or more rows and/or cells in CAM array 240. The data words are provided along a data bus 1618. An instruction decoder unit 1619 interprets instructions (e.g., write, read, compare) that are presented along an instruction bus 1621. The instruction decoder unit 1619 can also control the timing of any read, write or compare operations. If comparands used by the dynamic timing generator 220 are held by registers (not shown) associated with the dynamic timing generator 220, these comparands may be written/read to/from their registers via, for example, the data bus 1618. Here, the instruction decoder 1619 may be tailored to access these registers. Additionally, instruction decoder may provide one or more control signals

or timing signals to one or more of the other circuit blocks including provide CMPDSEL signals to dynamic timing generator 220.

[0075] Note that one or more of the address bus, data bus, instruction bus or bus carrying the comparand data may time multiplexed onto one or more of the other buses.

[0076] CAM array 240 may also be formed from several CAM array blocks each having their own address space as a subset of the entire address space for the CAM device. CAM arrays are conventionally blocked out to reduce signal line RC delays and speed the operation of the device. Here, each CAM array block may have its own dedicated dynamic timing generator, or may share a dynamic timing generator with one of the other CAM array blocks.

[0077] **Figure 17** illustrates one embodiment 1700 of an application (e.g., a networking application such as a switching card for a switch, a routing card for a router, or a line card for either or both a switch and/or a router) that can perform a look-up by using CAM device 1600 of **Figure 16**. A card is board-like structure having a plurality of semiconductor chips and the interconnections between them. Typically, the semiconductor chips are mounted to the card (e.g., by being soldered to the card or by being “plugged into” a socket that has been soldered to the card). The card embodiment of **Figure 17** includes a processor 1710, ingress circuitry 1730, egress circuitry 1740, CAM 1600, associated data storage unit 1770, traffic manager 1760, and payload storage unit 1750.

[0078] Processor 1710 functions to control the overall operation of the application. For example, processor 1710 receives packets from a network medium through ingress circuitry 1730, stores the payload of packets in payload storage unit 1750, and processes

packet header information to determine look-ups in CAM device 1600 and subsequent handling of the packets, as discussed herein. Ingress circuitry includes, for example, PHY and MAC devices. Processor 1710 sends out packets on a network medium through egress interface circuitry 1740 based on the lookups performed by CAM device 1600. Egress interface circuitry 1740 may be connected to a switch fabric or directly to one or more other routers or switches. Processor 1710 may be one or more network processor units (NPUs), microprocessors, or one or more special purpose processors such as a digital signal processor (DSP). In another embodiment, processor 1710 may be another type of controller, for example, a field programmable gate array or a general purpose processor. The processor 1710, ingress interface circuitry 1730, and egress interface circuitry 1740 components of a router are known in the art; accordingly, a detailed discussion is not provided.

[0079] In response to information in a packet header, for a particular packet, processor 1710 determines the number and types of lookups to be performed by one or more of CAM devices 1600, and forms the search keys for these lookups. The searches or lookups may include, for example, Classification lookups, forwarding lookups (e.g., Next Hop or longest prefix match (LPM) lookup, MAC lookup, MPLS lookup, etc.) When multiple searches are required, processor 1710 may form a composite search key that includes at least two, and as many as all, of the various search keys for the lookups. The composite search key may then be provided as a common input string to CAM device 1600. CAM device 1600 may selectively identify and extracts the individual search keys from the common input string and provides the individual search keys to associated CAM array blocks in CAM device 1600 to perform the lookups.

Advantageously, the lookups can then occur concurrently or simultaneously in the CAM array blocks of CAM device 1600, thereby increasing overall throughput over conventional systems in which searches are processed sequentially.

[0080] CAM device 1600 may be a multiple block CAM device with each block capable of storing a different table for comparand lookups. Alternatively, CAM device 1600 may represent multiple, single block CAM devices (e.g., with each single block CAM device formed on a different integrated circuit substrate) with each CAM device used to store a different table for comparand lookup. After one or more lookups are executed in CAM device 1600, associated information for matching entries (e.g., additional routing information and/or packet information) may be retrieved from associated data unit 1770. Processor 1710 then communicates with traffic manager 1760 to schedule the exit of a packet line card 1700 via egress interface circuitry 1740.

[0081] Note also that embodiments of the present description may be implemented not only within a semiconductor chip but also within machine readable media. For example, the designs discussed above may be stored upon and/or embedded within machine readable media associated with a design tool used for designing semiconductor devices. Examples include a circuit description formatted in the VHSIC Hardware Description Language (VHDL) language, Verilog language or SPICE language. Some circuit description examples include: a behavioral level description, a register transfer level (RTL) description, a gate level netlist and a transistor level netlist. Machine readable media may also include media having layout information such as a GDS-II file. Furthermore, netlist files or other machine readable media for semiconductor chip design

may be used in a simulation environment to perform the methods of the teachings described above.

[0082] Thus, it is also to be understood that embodiments of this invention may be used as or to support a software program executed upon some form of processing core (such as the Central Processing Unit (CPU) of a computer) or otherwise implemented or realized upon or within a machine readable medium. A machine readable medium includes any mechanism for storing or transmitting information in a form readable by a machine (e.g., a computer). For example, a machine readable medium includes read only memory (ROM); random access memory (RAM); magnetic disk storage media; optical storage media; flash memory devices; electrical, optical, acoustical or other form of propagated signals (e.g., carrier waves, infrared signals, digital signals, etc.); etc.

[0083] In the foregoing specification, the invention has been described with reference to specific exemplary embodiments thereof. It will, however, be evident that various modifications and changes may be made thereto without departing from the broader spirit and scope of the invention as set forth in the appended claims. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.